# Bessere Tickets.
# Klarere Entscheidungen.

Ein echtes Ticket prüfen. Nicht zutreffende Punkte begründen. Offene Punkte brauchen eine Frage, einen Owner und einen nächsten Schritt.

| TICKET / LINK | DECISION OWNER | REVIEW-DATUM |
| --- | --- | --- |
| | | |

## 01 Warum es wichtig ist

- [ ] Nutzerbedarf und angestrebtes Ergebnis sind explizit.
- [ ] Evidenz ist verlinkt; Annahmen und Evidenzgrenzen sind sichtbar.

## 02 Entscheidung und Grund

- [ ] Auswahl, Begründung und wichtiger Trade-off sind dokumentiert.
- [ ] Die Entscheidung hat Owner, Datum und eine stabile Referenz.

## 03 Scope und vereinbarte Referenz

- [ ] Enthaltene Arbeit, Ausschlüsse und relevante States sind klar.
- [ ] Release und Plattform verweisen auf eine exakte freigegebene Version.

## 04 Acceptance Checks

- [ ] Beobachtbare Checks sind mit relevanten Requirements verbunden.
- [ ] Quality Checks, Test Setup und Result Evidence sind bei Bedarf definiert.

## 05 Exceptions und Dependencies

- [ ] Abweichungen haben Grund, Limits und Follow-up - oder sind keine.
- [ ] Jede Dependency oder Exception hat Owner und Review-Trigger.

## 06 Review und offene Fragen

- [ ] Design, Product, Engineering und Testing haben beigetragen.
- [ ] Offene Fragen haben Owner; bei neuem Kontext erfolgt ein Review.

**OFFENE FRAGE / OWNER / NÄCHSTE AKTION / NICHT ZUTREFFEND: GRUND**

Design: Frage, Alternativen, Evidenz und Outputs prüfen. Development: Scope und prüfbares Verhalten untersuchen. Eine ausgefüllte Checklist bestätigt keine Delivery.

# Die Vereinbarung gemeinsam klären.

Ein 12-Minuten-Review für ein kommendes Ticket. Eine Person kann mehrere Perspektiven vertreten. Die Moderation dokumentiert im bestehenden Ticket.

**02 Min.**

### Separat lesen

Jede Person schreibt: Was wird umgesetzt? Warum? Was gilt als bestanden?

**03 Min.**

### Interpretationen vergleichen

Product: Ziel und Priorität. Design: Verhalten und Evidenz. Engineering: Dependencies und Machbarkeit. QA: States, Risiken und Checks.

**05 Min.**

### Lücken schließen

Referenz und Checks bestätigen. Trade-offs begründen. Bei Dissens Decision Owner und konkurrierende Annahmen benennen.

**02 Min.**

### Zurückerklären

Eine andere Person erklärt die Vereinbarung. Offene Fragen, Owner und Review-Trigger klären.

## Labels machen Review-Bedarf sichtbar

**ux-relevant**

Arbeit betrifft User Experience. Ticket Owner behält die Kategorie auch nach Abschluss.

**ux-review-needed**

Vereinbarung fehlt oder hat sich geändert. Ticket Owner entfernt es nach dokumentiertem Review, Vereinbarung und Datum.

**ux-exception**

Abweichung mit offenem Follow-up. Follow-up Owner entfernt es nach Lösung und dokumentiertem Abschluss.

**accessibility-review**

Konkrete Checks sind offen. Review Owner dokumentiert Checks, Ergebnisse und Defects vor dem Entfernen. Keine WCAG-Zertifizierung.

**WER PRÜFT DIE REVIEW QUEUE UND WANN?**

An drei Tickets pilotieren. Rückfragen, Suchzeit für Referenzen und Defects durch unterschiedliche Erwartungen beobachten. Gemeinsam prüfen, was hilft. Evaluationsideen, keine versprochenen Ergebnisse.

# Die Verbindung auffindbar machen.

Jira Cloud: Menüs hängen von Projekttyp, Berechtigung und Rollout ab. Atlassian nennt Projects teilweise "Spaces" und Issues "Work Items".

- [ ] Die sechs Überschriften von Seite 1 in Description einfügen und beantworten.
- [ ] Summary, Assignee und vereinbarte Labels in eigenen Jira-Feldern setzen.
- [ ] Design, Development und Follow-ups verlinken; Blocking Links nur für echte Dependencies.
- [ ] Query prüfen, Suche speichern und einen Review Queue Owner benennen.

**UX Review offen / DEMO durch euren Project Key ersetzen**

```
project = "DEMO" AND
labels = "ux-review-needed" AND
statusCategory != Done ORDER BY updated DESC
```

Done-Category ausgeschlossen. Bei länger offenen Exceptions einen offenen gelabelten Follow-up halten. Filter-Sharing gibt keinen Ticketzugriff.

## Setup passend zum Projekttyp wählen

### TEAM-MANAGED ADMIN

- [ ] Board/Backlog > ... > Manage custom filters. Label Query verwenden.
- [ ] Optional: Space settings > Fields. Lokale Fields bleiben im Projekt.

### COMPANY-MANAGED ADMIN

- [ ] Manage custom filters oder Configure board > Quick Filters. Begrenzt vorhandene Board-Ergebnisse.
- [ ] Description und Referenz sichtbar halten. Zusätzliche Fields/Screens durch Jira Admin.

Optional: UX review state (Not reviewed / Review needed / Reviewed) plus Datum. Bedeutung und Pflege vereinbaren; doppelte States vermeiden.

### Quellen / genaue Anleitungen öffnen

Jira: Linked Work / Saved Searches

Filter: team-managed / company-managed | Fields

Decision Records: Nygard / IREB Handbooks / Interaktives Writing + Ticket Builder

Vorgeschlagene Praxis von Pegah Shademani. Labels sind Vorschläge, keine Jira Defaults. Stand: 7. Oktober 2026. Die Checklist gibt keine Entscheidung frei und zertifiziert keine Accessibility. Digital ausfüllen oder drucken.